Commonsense about Human Senses: Labeled Data Collection Processes

Ndapa Nakashole
Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093
nnakashole@eng.ucsd.edu

Abstract

We consider the problem of extracting from text commonsense knowledge pertaining to human senses such as sound and smell. First, we consider the problem of recognizing mentions of human senses in text. Our contribution is a method for acquiring labeled data. Experiments show the effectiveness of our proposed data labeling approach when used with standard machine learning models on the task of sense recognition in text. Second, we propose to extract novel, common sense relationships pertaining to sense perception concepts. Our contribution is a process for generating labeled data by leveraging large corpora and crowdsourcing questionnaires.

1 Introduction

Information extraction methods produce structured data in the form of knowledge bases of factual assertions. Such knowledge bases are useful for porting inference, question answering, and reasoning (Bollacker et al., 2008; Hoffart et al., 2012; Mitchell et al., 2015). However, progress on the common sense front, as opposed to named entities such as locations, and people, is still limited (Havasi et al., 2007; Tandon et al., 2011).

One of the factors impeding progress in common sense knowledge acquisition is the lack of labeled data. Prior work has shown that it can be straightforward to obtain training data for identifying relationships between named entities such as companies and their headquarters, or people and their birth places (Havasi et al., 2007; Tandon et al., 2011; Bollacker et al., 2008; Hoffart et al., 2012; Mitchell et al., 2015). Examples of such relationships can be found in semi-structured formats on the Web(Wu and Weld, 2008; Wang and Cohen, 2008). This is not the case for common sense relationships.

We therefore consider the problem of extracting from text commonsense knowledge pertaining to human senses such as sound and smell. We split the problem into two parts, for each part we propose approaches for obtaining labeled data, and train standard machine learning models.

1. In the first part of this work, the goal is to detect mentions of concepts that are discernible by sense. For example, recognize that "chirping birds" is a mention of an audible concept (sound), and "burning rubber" is a mention of an olfactible concept (smell). We aim to detect *mentions* of concepts without performing co-reference resolution or clustering mentions. Therefore, our setting resembles the established task of entity recognition (Finkel et al., 2005; Ratinov and Roth, 2009), with the difference being that we focus on un-named entities.

   We propose a data labeling method, that leverages crowd-sourcing and large corpora. This approach provides the flexibility to control the size and accuracy of the available labeled data for model training. Additionally, we train several standard machine learning models including to recognize mentions of sound and smell concepts in text. In our experiments, we show that the combination of our data labeling approach, and a suitable learning model are an effective solution to sense recognition in text.

2. In the second part of this work, we seek to extract novel common sense relationships about concepts that are discernible by sense.

   Our contributions in this part of the work are as follows: first, we propose to extract novel relationships that are sparse in existing knowledge bases. Second, we propose a pro-

Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing, pages 43–52
Hongkong, China, November 3, 2019. ©2019 Association for Computational Linguistics

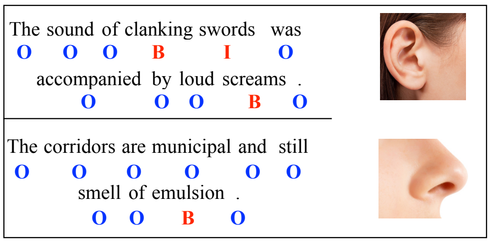
The sound of clanking swords was
O O O B I O
accompanied by loud screams .
O O O B O
The corridors are municipal and still
O O O O O O
smell of emulsion .
O O B O

Figure 1: Example beginning-inside-outside (BIO) labeled sentences with mentions of sound (top) and smell (bottom) concepts.

cess for generating labeled data by leveraging large corpora and crowd-sourcing questionnaires. Third, using the resulting labeled data, we train standard machine learning methods (both linear model and memory neural network models), obtaining high accuracy on the task of extracting these previously under-explored relationships.

In summary, we propose minimal-effort approaches for obtaining labeled data on two key tasks: mention recognition, and relationship extraction for concepts pertaining to human senses. In the first, task we make use of Hearst patterns, and crowd sourcing, and for the second task, we make use of part-of-speech tag sequences and crowd-sourcing. Although these processes are not new, we have applied them to a novel setting of common sense about human senses, and showed their effectiveness. We trained standard machine learning methods, and showed that the labeled data generated by our processes lead to high quality models.

2 Recognizing Mentions of Human Senses

In this part of the work, we would like to detect mentions of concepts discernible by sense, we focus on mentions of *audible (sound)* and *olfactible (smell)* concepts. We treat sense recognition in text as a sequence labeling task where each sentence is a sequence of tokens labeled using the BIO tagging scheme (Ratinov and Roth, 2009). The BIO labels denote tokens at the *beginning, inside, and outside* of a relevant mention, respectively. Example BIO tagged sentences are shown in Figure 1.

2.1 Data Labeling Methodologies

There is a lack of easy to identify labeled data on the Web for common sense information extraction, an issue which affects named-entity centric information extraction to a lesser degree (Wang and Cohen, 2008; Wu and Weld, 2008). We consider three data labeling approaches: *i*) Automatically generate training data using judiciously specified patterns. *ii)* Solicit input on crowd-sourcing platforms. *iii)* Leverage both i) and ii) in order to overcome their respective limitations.

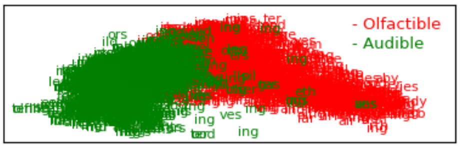

Figure 2: A PCA projection of the embeddings of audible and olfactible phrases labeled by the pattern approach.

2.1.1 Pattern-based Corpus Labeling

To label data with patterns, we begin by specifying patterns that we apply to a large corpus. For our concepts of interest, sound, and smell, we specify the following two patterns. " sound of <y>", and " smell of <y>", We then apply these patterns to a large corpus. In our experiments, we used the English part of ClueWeb09. [1]. The result is a large collection of occurrences such as: " sound of *breaking glass*", "smell of *perfume*", etc. The collections contains 134,473 sound phrases, and 18,183 smell phrases.

Figure 2, shows a 2D projection of the 300-dimensional word vectors[2] of the discovered audible and olfactible phrases. We see a strong hint of two clusters. We later provide a quantitative analysis of this data.

2.1.2 Crowd-Sourced Supervision

The second way of obtaining labeled data that we consider is crowd-sourcing. We used the Amazon Mechanical Turk crowd-sourcing platform.

Crowd Task Definition. To obtain labeled examples, we could do a "cold call" and ask crowd workers to list examples of phrases that refer to senses. However, such an approach requires crowd workers to think of examples without clues or memory triggers. This is time consuming and error prone. We propose to exploit a large corpus to obtain preliminary labeled data, making it possible to only need crowd workers to filter the data through a series of *"yes/no/notsure"* questions. These types of questions require little effort

[1] http://lemurproject.org/clueweb09/

[2] https://code.google.com/archive/p/word2vec/

	% Majority Yes	*Fleiss* κ
Audible	73.4%	0.51
Olfactible	89.6%	0.33

Table 1: Crowd-sourced labeling of phrases generated by the pattern approach of section 2.1.1.

from crowd workers while mitigating the amount of noisy input that one could get from open-ended questions. We randomly selected 1000 phrases labeled by the pattern approach as described in Section 2.1.1 to be sound/smell phrases, 500 for each sense type. Each phrase was given to 3 different workers to annotate *"yes/no/notsure"*. We consider a phrase to be a true mention of the labeled sense if the majority of the participants chose "yes". This annotation task serves two purposes: 1) to provide us with human labeled examples of sound and smell concepts ii) to provide a quantitative evaluation of pattern generated labels.

Crowd Annotation Results. Table 1 is a summary of the annotation results. First, we can see that the accuracy of the patterns is quite high, which was hinted at in Figure 2. Second, The inter-annotator agreement rates are moderate, but lower for olfactible phrases. This is also reflected by the fact that there were around 3 times as many "not sure" responses in the smell annotations as there were in the sound annotation task (27 vs 10). Nonetheless, the output of these tasks provide us with another option for labeled data that we can use to train our models.

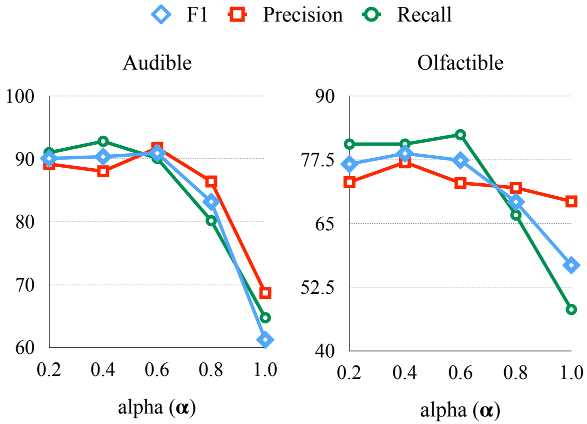

Figure 3: Performance as α is varied to control size and accuracy of labeled data.

2.1.3 Joint Pattern & Crowd-Sourced Labeling

A third way of obtaining labeled data is to leverage both pattern-based and crowd-sourced labeling approaches. One central question pertains to how we can combine the two sources in a way that exploits the advantages of each approach while mitigating their limitations. We seek to start with the crowd-sourced labeled, which is small but more accurate, and expand it with the pattern-generated labeled data, which is large but less accurate. We define a function that determines how to expand the data. Let $x_i^c \in D^c$ be a crowd labeled phrase, and $x_i^p \in D^p$ be a pattern labeled phrase. Then x_i^p is added to our training labeled data D^{pc} if $sim(x_i^c, x_i^p) >= \alpha$ where sim is the cosine similarity between the vector representations of the phrases. For vector representations of phrases, we use the same pre-trained Google word embeddings as those used to plot Figure 2. For phrases longer than one word, we use vector averaging. The effect of varying α, for a fixed prediction model, can be seen in Figure 3. When $\alpha = 1$, that is we are only using the crowd-sourced labeled data, performance is at its worst. This is because even though the human labeled data is more accurate, it is much smaller, leading to potential model overfitting problems. A more subtle finding is that with low α values (i.e., <0.4 for audible concepts), we have the highest recall, but not the best precision, this can be explained by the fact that, with low α values, we are allowing more of the automatically labeled data to be part of the training data, thereby potentially adding noise to the model. However, the advantage of the mixture approach comes from the fact that, there comes a point where precision goes up, recall slightly degrades but we obtain the best F1 score. In Figure 3, we see these points at $\alpha = 0.6$ and $\alpha = 0.4$ for the audible and olfactible concepts respectively. We use these values to generate the labeled data used to train models described in the rest of the paper.

2.2 Learning Models

We treat sense recognition in text as sequence prediction problem, we would like to estimate: $P(y_i|x_{i-k},...,x_{i+k};y_{i-l},...,y_{i-1})$. where x refers to words, and y refers to BIO labels.

Conditional Random Fields (CRFs) (Lafferty et al., 2001) have been widely used named entity recognition (Ratinov and Roth, 2009; Finkel et al., 2005), a task similar to our own. While the CRF models performed reasonably well on our task, we sought to obtain improvements by training variations of Long Short Memory (LSTM) re-

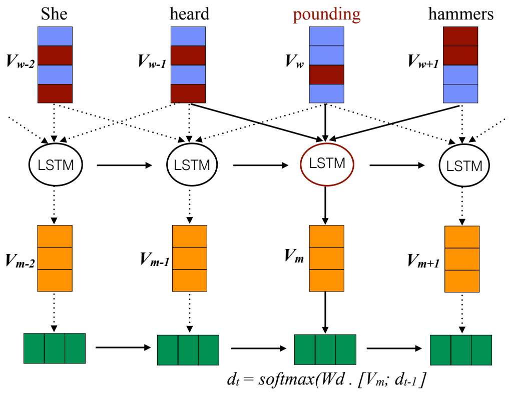

Figure 4: Our neural network architecture for the task of recognizing concepts that are discernible by sensesss.

current neural networks (Hochreiter and Schmidhuber, 1997). We found variations of LSTM sequence classifiers to do better than the CRF model, and also better than standard LSTMs. In particular, the well-studied combination of CRF and LSTMs works better.

Word and Character Features. As input, the LSTM neural network model takes a sentence and, as output, produces a probability distribution over the BIO tags for each word in the sentence. To BIO tag each word in the sentence, we use word features. We chose the word features to be their word embeddings. As additional features, we model the character composition of words in order to capture morphology. Neural encodings of character-level features have been shown to yield performance gains in natural language tasks (Ling et al., 2015; Chiu and Nichols, 2016). In all our experiments, we initialize the word embeddings with the Google news pre-trained word embeddings [3]. The character embeddings are learned from scratch.

Prediction and Output Layer Recurrence. We represent each word as a mention within a short context window of length m. We use the LSTM to encode these windows contexts in order to make a prediction for each word. The LSTM window encoding is then used to make predictions over the BIO labels. The output for each word is decoded by a linear layer and a *softmax* layer into probabilities over the BIO tag labels. Crucially, we modify the standard LSTM by modeling temporal dependencies by introducing a recurrence in the output layer. Therefore, the prediction $\boldsymbol{d}_t$ at time step t takes into account the prediction $\boldsymbol{d}_{t-1}$ at the previous time t-1. Formally, we have: $\boldsymbol{d}_t = \text{softmax}(\boldsymbol{W}_d \cdot [\boldsymbol{v}_m; \boldsymbol{v}_{c_a}; \boldsymbol{v}_s; \boldsymbol{d}_{t-1}])$, where $\text{softmax}(z_i) = e^{z_i} / \sum_j e^{z_j}$. We illustrate the model in Figure 4. We found this model to consistently perform well on the senses of sound and smell.

Sound	**Smell**
honking cars	burning rubber
snoring	chlorine
gunshots	citrus blossoms
live music	fresh paint

Table 2: Examples of sound and smell concepts recognized by our method.

Model Evaluation. To evaluate the models, we set aside 200 of the 1000 crowd-annotated phrases as test data, meaning we have 100 test instances for each sense type (sound/smell). The rest of the data, 400 per sense type was used for generating training data using the combined crowd and pattern approach described in Section 2.1.3. We set $\alpha = 0.6$ and $\alpha = 0.4$, based on Figure 3, for audible and olfactible concepts respectively. With these α values, the combination approach produced *1,962* and *1,702* training instances for audible and olfactible concepts respectively

Performance of the various models is shown in Table 3. The abbreviations denote the following: **LSTM** refers to a vanilla LSTM model, using only word embeddings as features, **+ OR** refers to the LSTM plus the output recurrence, **+ CHAR** refers to the LSTM plus the character embeddings as features. **+ OR + CHAR** refers to the LSTM plus the output recurrence and character embeddings as features. For the **CRF**, we use the commonly used features for named entity recognition: words, prefix/suffices, and part-of-speech tag (Ratinov and Roth, 2009). We can see that for both senses, the model that uses both character embedding features, and an output recurrence layer yields the best F1 score. Examples of sounds and smells our method can recognize are shown in Table 2.

2.3 Sense Mention Recognition Related Work

Our task is related to entity recognition however in this paper we focused on novel types of entities, which can be used to improve extraction of common sense knowledge. Entity recognition systems are traditionally based on a sequen-

[3] https://code.google.com/p/word2vec/

Method	F1	P	R
	Audible		
CRF	89.38	87.83	90.99
LSTM	89.64	88.87	90.42
+ OR	89.780	88.60	90.99
+CHAR	87.78	88.18	87.39
+ OR + CHAR	**90.91**	91.740	90.09
	Olfactible		
CRF	75.73	79.59	72.22
LSTM	69.96	62.96	78.70
+ OR	78.380	76.320	80.56
+ CHAR	69.57	60.69	81.48
+ OR + CHAR	**78.73**	76.990	80.56

Table 3: Performance of the various models on the task of sense recognition.

tial model, for example a CRF, and involve feature engineering (Lafferty et al., 2001; Ratinov and Roth, 2009). Like other neural approaches, our approach does not require feature engineering (Hammerton, 2003; Collobert et al., 2011; dos Santos and Guimarães, 2015; Chiu and Nichols, 2016; Shimaoka et al., 2016), the only features we use are word and character embeddings. The work of (Lample et al., 2016) introduced a CRF on top of LSTM for the task of named entity recognition.

2.4 Summary on Sense Mention Recognition

We have presented a method for recognizing concepts that are discernible by sense by proposing a process for collecting data, and then training standard machine learning methods. The concepts our method recognizes present opportunities for discovering additional types of common sense knowledge, for example, learning relationships that encode information such as which objects produce which sounds, in which environments can certain sounds be found, what is the sentiment of various types of smell, etc. These type of relations can significantly improve coverage of common sense in knowledge bases, thereby improving their utility. We explore this direction in the next section.

3 Relationships for Concepts Discernible By Sense

Now that we have a way of recognizing mentions of sense concepts in text, we can move on to relationships between such concepts.

To focus our task, we consider three relations pertaining to sense perception of sound and smell. Namely: 1) *soundSourceRelation*, 2) *soundSceneRelation*, and 3) *smellSentimentRelation*.

3.1 Sound-Source Relationship

The sound-source relationship represents information pertaining to which objects produce which sounds. For example, that planes and birds are capable of *flying*, the wind *blows*, and geckos *bark*. Obtaining sufficient labeled data to learn an extractor for this relationship is non-trivial, we propose one approach in the next section.

Labeled Data Generation. One option for obtaining labeled data is to directly request for it on crowd-sourcing platform by asking crowd workers to list examples of sounds and their sources. However, such an approach requires crowd workers to think of examples without clues or memory triggers. This is time consuming and error prone. Therefore, as we did in the recognition task, we again propose to exploit a large corpus to obtain preliminary labeled data. This way, we again only need crowd workers to filter the data through a series of *"yes/no/notsure"* questions. These type of questions require little effort from crowd workers while mitigating the amount of noisy input that one could get from open-ended questions.

To pose *"yes/no/notsure"* questions, we need a list of plausible sound-source pairs. To this end, we propose a lightly supervised corpus-based technique. First, we identify which phrases refer to sounds using the approach described in the first Section 2

One important observation we made was that about 20,000 (15%) of the 134,471 phrases are bigrams of the form: "verb noun" or "noun verb" where in both cases, the verb is in the gerund or present participle V-*ing* form. For example, *birds chirping, cars honking,squealing brakes*, etc. From phrases of this kind, we create verb-noun pairs, that we treat as plausible sound-source pairs where the verb is the *sound* and the noun is the *source*. We then asked crowd-workers to decide if the source (noun) produces the sound (verb). Thus from "birds chirping" we generate the question, "Is *chirping* a sound produced by *birds*?"; Negative examples include: "surrounding nature", and "Standing ovation", i.e., standing is not a sound made by ovation. We generated 634 such questions, from which we obtained a moderate inter-

	Fleiss κ
soundSource	0.57
soundEnvironment	0.35
smellSentiment	0.43

Table 4: Fleiss κ. inter-annotator agreement rates for the three relations on yes/no type crowd-sourcing tasks.

annotator agreement rate of Fleiss $\kappa = 0.57$, see Table 4. We use the resulting labeled data to train two types of learning methods.

Linear Learning Model. The learning problem for the sound-source relationship is as follows: given a bi-gram phrase n of the form "verb noun" or "noun verb", we wish to classify yes or no if a given noun, denoted by w_{src}, produces the verb, denoted by word w_{snd}, as a sound. As a simple linear solution to this problem, we train a logistic regression classifier. The features we use are the vectors representing the word embeddings of w_{src} and w_{snd}, denoted by $\boldsymbol{v}_{src}$, and $\boldsymbol{v}_{snd}$. In our experiments, we use the 300-dimensional Google News pre-trained embeddings [4]. There are several ways in which we combine $\boldsymbol{v}_{src}$, and $\boldsymbol{v}_{snd}$ into a single feature vector:

Vector Concatenation: $v = concat(\boldsymbol{v}_{src}, \boldsymbol{v}_{snd})$
Size of v, $|v| = |\boldsymbol{v}_{src}| + |\boldsymbol{v}_{snd}|$
LSTM encoder : $v = lstm(\boldsymbol{v}_{src}, \boldsymbol{v}_{snd})$
An LSTM (Hochreiter and Schmidhuber, 1997) recurrent neural network is used to encode the phrase containing $\boldsymbol{v}_{src}$ and $\boldsymbol{v}_{snd}$. $|v| = h$, where h is the hidden layer size of the neural network.
Source minus sound: $v = \boldsymbol{v}_{src} - \boldsymbol{v}_{snd}$
$|v| = |\boldsymbol{v}_{src}| = |\boldsymbol{v}_{snd}|$
Sound minus source: $v = \boldsymbol{v}_{snd} - \boldsymbol{v}_{src}$
$|v| = |\boldsymbol{v}_{src}| = |\boldsymbol{v}_{snd}|$

Memory Networks Learning Model. In addition to the variations of the linear model, we also trained a non-linear model in the form of memory networks (Sukhbaatar et al., 2015). Memory networks combine their inference component with a memory component. The memory component serves as a knowledge base or history vault to recall words or facts from the past. For the task of relation extraction, the memory network model learns a scoring function to rank relevant memories (words) with respect to how much they express a given relationship. This is done for a given argument pair as a query, i.e., a sound-source pair. At prediction time, the model finds k relevant memories (words) according to the scoring function and conditions its output on these memories. In our experiments, we explore different values of k, effectively changing how many memories (words), the model conditions on. We report results for up to $k = 3$ as we did not see improvements for larger values of k.

Learning Model	**Accuracy**
LM: LSTM encoder	**0.90**
LM: (Source - Target)	0.88
LM: (Target - Source)	0.87
LM: Vector Concatenation	0.83
MM: 1 hop	0.87
MM: 3 hops	0.85

Table 5: Accuracy of the linear models (LM) and memory networks models (MM) on the sound-source relation.

Sound-Source Evaluation. Both the linear model and the memory networks models were implemented using Tensorflow. For the memory networks, we implemented the end-to-end version as described in (Weston et al., 2014; Sukhbaatar et al., 2015). Of the 634 crowd-sourced labeled examples described, we used 100 as test data, the rest as training data. Model parameters such as hidden layer size of the memory networks were tuned using cross-validation on the training data. As shown in Table 2, we obtain high accuracy across all models. The best performing model is a linear model with an LSTM encoding of the sound phrases, achieving accuracy of 90%. Surprisingly, we could not obtain better results with the memory networks model. Increasing the memory size or the number of hops (how often we iterate over the memories) did not help. One possible reason is the size of our training data, in previous work (Weston et al., 2014; Sukhbaatar et al., 2015), the memory networks were trained on 1,000 or more examples per problem type whereas our training data is half the size. Nevertheless, the memory networks module still produces good accuracy, with best performance of 87%.

3.2 Sound-Scene Relationship

The sound-scene relationship represents information about which sounds are found in which scenes. For example, birds chirping can be found in a forest. Therefore, this kind of information can

[4] https://code.google.com/archive/p/word2vec/

also be used in context recognition systems (Eronen et al., 2006), in addition to providing common sense knowledge that could be useful in language understanding tasks.

Labeled Data Generation. We would like to obtain labeled data in the form of scenes and their sounds. For example, (beach, waves crashing), (construction, hammering), (street, sirens), (street, honking cars). To obtain this type of labeled data, we again would like to only use *"yes/no/notsure"* crowd-sourcing questions. To generate plausible sound-scene pairs, first we find all sentences that mention at least one scene and one sound concept. To detect sound concepts, we use the approach described in Section 2. To detect mentions of scenes, we specified a list of 36 example scenes, which includes scenes such as beach, park, airport most of our scenes are part of the list of acoustic scenes from a scene classification challenge [5]. For every sentence that mentions both an acoustic scene and a sound concept, we apply a dependency parser[6]. This step produces dependencies that form a directed graph, with words being nodes and dependencies being edges.

Dependency graph shortest paths between entities have been found to be a good indicator of relationships between entities (Xu et al., 2015; Nakashole et al., 2013b). We use shortest paths as features in order classify sound-scene pairs. To obtain training data, we sort the paths by frequency, that is, how often we have seen the path occur with different sound-scene pairs. We then consider pairs that occur with frequent shortest paths to be plausible sound-scene pairs which we can present to crowd-workers in *"yes/no/notsure"* questions. We randomly selected 584 sound-scene pairs, and the corresponding sentences that mention them, which were then presented to crowd workers in questions. The inter-annotator agreement rate on this task is Fleiss $\kappa = 0.35$, see Table 4.

Learning Models and Evaluation. For the linear model, we consider three options for features. *Shortest Paths (SP)*: LSTM encoding of the dependency shortest path. *Sentence (S)*: an LSTM encoding of the sentence. *SP + S*: encoding of both the shortest path and the sentence are used as features. For the memory network models, we considered using the contents of both the shortest paths and the sentences to produce memories. We use 100 of the 584 labeled data for testing, the rest for training. The shortest paths performed better, for space reasons we omit the results of using sentences as memories. As shown in Table 6, the linear model with the shortest path achieves the best accuracy of 81%. However, the best performing memory networks model with 3 memory hops is not significantly worse at 80% accuracy.

Learning Model	Accuracy
LM: shortest path	**0.81**
LM: shortest path +sentence:	0.80
LM: sentence	0.75
MM: 1 hop	0.75
MM: 3 hops	0.80

Table 6: Accuracy on the sound-scene relation.

Learning Model	Accuracy
LM: LSTM encoder	**0.84**
LM: vector addition	0.81
MM: 1 hop	0.82
MM: 3 hops	0.82

Table 7: Accuracy on the sound-sentiment relation.

3.3 Smell-Sentiment Relationship

For the smell-sentiment relationship, the goal is to extract information about which smells are considered pleasant, unpleasant or neutral. In general, sentiment is both subjective and context dependent. However, as we show through crowd-sourced annotations, there is substantial consensus even on sentiment of smells.

Labeled Data Generation. First we generate a list of plausible smells phrases, following a similar approach to Section 2. We then used these phrases to evaluate sentiment of smells in a Mechanical Turk task. We present a phrase within a sentence context. We then asked crowd workers to choose if the phrase refers to a smell that is *"pleasant/unpleasant/neutral/notsure/notasmell"*. We generated 600 such questions on which we obtained a moderate inter-annotator agreement rate of Fleiss $\kappa = 0.43$, see Table 4. While this is not a yes/no task, it is still a simple multiple choice task with the same advantages of the yes/no tasks as we described earlier.

Learning Models and Evaluation. We again use the same earning models. For the linear model,

[5] http://www.cs.tut.fi/sgn/arg/dcase2016/

[6] https://pypi.python.org/pypi/practnlptools/1.0

we consider two options for features. **LSTM encoder**: LSTM encoding of the smell phrase **Vector addition**: vector addition encoding of the smell phrase. For the memory network models, the contents of the sentence that mentions the phrases are stored as memories. We use 100 of the 600 labeled data for testing, the rest for training. As can be seen in Table 7, the linear model with LSTM encoded phrases achieved the highest accuracy of 84%.

3.4 Summary on Relationships

In this work, we extracted novel common sense relations, using standard machine learning methods. To obtain labeled data, we proposed a combination of large corpora, and multiple choice crowd-sourced questions. These type of questions require little effort from crowd workers while mitigating the amount of noise one might get from open-ended questions. We have also proposed and trained models on this data, achieving high accuracy for all relations. Scaling up our approach to more relations is an exciting future direction for our work. Scale is not expected to be prohibitive, given the minimally-supervised nature of our approach.

4 Conclusion

Cyc (Lenat, 1995), and ConceptNet (Havasi et al., 2007) are well-known examples of knowledge bases of everyday common sense knowledge. These projects are decades long efforts involving either experts or crowd-sourcing. Other knowledge bases focus on facts about named entities such as people, locations, and companies (Bollacker et al., 2008; Hoffart et al., 2012; Mitchell et al., 2015). Common sense contained in these knowledge bases is still limited . We considered the problem of extracting from text commonsense knowledge pertaining to human senses such as sound and smell. We proposed minimal-effort approaches for obtaining labeled data on two key tasks: mention recognition, and relationship extraction. In the first task we make use of Hearst patterns, and crowd sourcing, and for the second task, we make use of part-of-speech tag sequences and crowd-sourcing. Although these processes are not new, we have applied them to a novel setting of common sense about human senses, and showed their effectiveness. We trained standard machine learning methods, and showed that the labeled data generated by our processes lead to high quality models.

In the future, we would like to apply our methods to a broader class of common sense assertion, and to go develop novel machine learning methods that improve accuracy on both of these tasks.

References

Yoshua Bengio, P. Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks. Special Issue on Recur.*

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*, SIGMOD '08, pages 1247–1250.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *WebDB*, pages 172–183.

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *TACL*, 4:357–370.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Antti J Eronen, Vesa T Peltonen, Juha T Tuomi, Anssi P Klapuri, Seppo Fagerlund, Timo Sorsa, Gaëtan Lorho, and Jyri Huopaniemi. 2006. Audio-based context recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14:321–329.

Christaine Fellbaum. 1998. A semantic network of English verbs. In *WordNet: An Electronic Lexical Database*, pages 69–104. The MIT Press.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*.

James Hammerton. 2003. Named entity recognition with long short-term memory. In *HLT-NAACL*, pages 172–175.

Catherine Havasi, Robert Speer, and Jason Alonso. 2007. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *RANLP*, pages 27–29.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING*, pages 539–545.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(1):1–42.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2012. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 782–792.

Anurag Kumar, Bhiksha Raj, and Ndapandula Nakashole. 2017. Discovering sound concepts and acoustic relations in text. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 631–635. IEEE.

Matthieu Labeau, Kevin Löser, and Alexandre Allauzen. 2015. Non-lexical architecture for fine-grained POS tagging. In *EMNLP, 2015*, pages 232–237.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270.

Douglas B. Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11).

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luís Marujo, and Tiago Luís. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *EMNLP*, pages 1520–1530.

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending learning. In *AAAI*, pages 2302–2310.

Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013a. Fine-grained semantic typing of emerging entities. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL*, pages 1488–1497.

Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. 2013b. Discovering semantic relations from the web and organizing them with PATTY. *SIGMOD Record*, 42(2):29–34.

Ndapandula T Nakashole. 2012. Automatic extraction of facts, relations, and entities for web-scale knowledge base population.

Lev-Arie Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*, pages 147–155.

Cícero Nogueira dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. *CoRR*, abs/1505.05008.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An attentive neural architecture for fine-grained entity type classification. *arXiv preprint arXiv:1604.05525*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448.

Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,*, pages 57–66.

Niket Tandon, Gerard de Melo, and Gerhard Weikum. 2011. Deriving a web-scale common sense fact database. In *AAAI*.

Niket Tandon, Gerard de Melo, and Gerhard Weikum. 2014. Acquiring comparative commonsense knowledge from the web. In *AAAI*, pages 166–172.

Richard C. Wang and William W. Cohen. 2008. Iterative set expansion of named entities using the web. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 1091–1096.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint https://arxiv.org/abs/1410.3916*.

Derry Tanti Wijaya, Ndapandula Nakashole, and Tom Mitchell. 2015. “a spousal relation begins with a deletion of engage and ends with an addition of divorce”: Learning state changing verbs from wikipedia revision history. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 518–523.

Fei Wu and Daniel S. Weld. 2008. Automatically refining the wikipedia infobox ontology. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pages 635–644.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1785–1794.